CLAIMS

What is claimed is:

1. A method of maintaining clock consistency in a fault-tolerant distributed system for a group of replicas having physical hardware clocks, comprising:

executing a time service handler for accessing a physical hardware clock;

establishing a single consistent group clock value within said time service handler for a group of replicas to be used in place of said physical hardware clock value; and

returning from said time service handler with said group clock value.

2. A method as recited in claim 1, wherein said replicas in said group receive a single consistent group clock value while at least one of said replicas continues to operate.

3. A method as recited in claim 1, wherein in response to successive executions of said time service handler by said replicas in said group for accessing said physical hardware clocks, the consistent group clock values returned to said replicas in said group are monotonically increasing.

4. A method as recited in claim 1, wherein in response to successive executions of said time service handler by said replicas in said group, the increment, skew, and drift of the consistent group clock value from one reading to the next is bounded.

5. A method as recited in claim 1:

wherein said time service handler computes and stores a clock offset value as the difference between the physical hardware clock value and the most recent determination of said consistent group clock value;

wherein said clock offset value is updated in response to establishing a consistent group clock value.

6. A method as recited in claim 5, wherein a replica determines a logical local clock value as a proposed consistent group clock value for communication to other replicas in said group of replicas, by adding said clock offset value to said physical hardware clock value read from hardware.

7. A method as recited in claim 6, wherein said proposed consistent group clock value is communicated to other replicas of said group by multicasting said proposed consistent group clock value to all of the replicas in said group.

8. A method as recited in claim 7, wherein said multicasting is performed according to a reliable ordered multicast protocol which ensures that all replicas receive the same messages in the same order, and that a message is either delivered to all of the replicas in said group or to none of them.

9. A method as recited in claim 7, wherein the first consistent clock synchronization message being multicast contains a proposed consistent group clock value for that reading equal to the physical hardware clock value for use by other replicas in said group of replicas as said consistent group clock value for that reading.

10. A method as recited in claim 6:

further comprising joining said group of replicas by a new replica or repaired replica which receives consistent clock synchronization information upon joining;

wherein said new or repaired replica can subsequently participate in the determination of group clock values.

11. A method as recited in claim 1, wherein said establishing of a single consistent group clock value within said time service handler comprises:

reading a physical clock value in response to a clock operation performed by said time service handler for a thread of a given replica;

determining a local logical clock value for said given replica by adding a clock offset value to said physical clock value;

sending a clock synchronization message to replicas in said group of replicas, which proposes said local logical clock value as a consistent group clock value, when no other clock synchronization messages for the calling thread have been received;

extracting said local clock value from a received clock synchronization message as a consistent group clock value; and

updating said clock offset value for said given replica to the value of the consistent group clock value less the physical clock value.

12. A method of maintaining clock consistency for a group of replicas having physical hardware clocks and operating in a fault-tolerant distributed system, comprising:

reading a physical clock value for a given replica in response to a clock operation for a thread;

determining a local logical clock value for said given replica by adding a clock offset value to said physical clock value;

proposing said local logical clock value as a group clock value within a clock synchronization message sent to replicas in said group of replicas when no other clock synchronization messages for the calling thread for this reading of the physical clock value have been received;

extracting said local logical clock value from a received clock synchronization message which is established as a group clock value that is consistent across said group of replicas;

updating said clock offset value for said given replica to the value of the group clock value less the physical clock value; and

returning said group clock value to said thread of said given replica.

13. A method as recited in claim 12, wherein the succession of said group clock values established for said thread of said given replica is monotonically increasing.

14. A method as recited in claim 12, wherein said clock synchronization message includes a common fault-tolerant protocol message header.

15. A method as recited in claim 12, wherein said clock synchronization message is sent to replicas of said group by multicasting according to a reliable ordered multicast protocol which ensures that all replicas receive the same messages in the same order, and that a message is delivered either to all of the replicas in the group or to none of them.

16. A method as recited in claim 12, wherein said process of reading, determining, proposing, extracting, and updating is performed within a time service handler which is executed in response to said clock operation for said thread of said given replica.

17. A method as recited in claim 12, wherein said extracted local logical clock value may be received from said given replica if a clock synchronization message from other replicas is not received first.

18. A method as recited in claim 12, further comprising initializing said clock offset value upon commencing execution of each replica in said group of replicas.

19. A method as recited in claim 12, wherein said extracting of said local logical clock value from said clock synchronization message comprises:

extracting a sending thread id from said clock synchronization message;

queuing said clock synchronization message in an input buffer associated with said thread id;

extracting said local logical clock value from said clock synchronization message; and

setting said group clock value to said local logical clock value.

20. A method as recited in claim 12, wherein said local logical clock value proposed within said clock synchronization message as sent by said primary replica establishes said group clock value when executing passive and semi-active replication strategies.

21. A method as recited in claim 12, wherein said local logical clock value proposed within said clock synchronization message is established as said group clock value in response to the order that the clock synchronization message is sent to replicas in said group when executing active replication strategies.

22. A method as recited in claim 12, wherein the process of reading, determining, proposing, extracting, and updating is performed in successive rounds identified by a round number, to present a consistent view of the clock for the replicas in the group.

23. A method as recited in claim 22, wherein said clock synchronization message contains a proposed group clock value, group identifier, thread identifier, and said round number.

24. A method as recited in claim 23, wherein said thread executing said clock operation is blocked waiting for the arrival of a first matching clock synchronization message.

25. A method as recited in claim 24, wherein said clock synchronization message for said thread matches if it has the same round number.

26. A method as recited in claim 22, wherein a new round of clock synchronization is started for each clock-related operation.

27. A method as recited in claim 26, further comprising initializing a round number value upon commencing replica execution.

28. A method as recited in claim 26, wherein said round number is utilized for detecting duplicate clock synchronization messages under active replication strategies.

29. A method as recited in claim 26, wherein said round number is utilized for matching the clock-related operation of a thread with the corresponding clock synchronization message.

30. A method as recited in claim 26, wherein said proposed local logical clock value of a synchronizer replica is selected for a given round to determine the group clock value for said group of replicas.

31. A method as recited in claim 30, wherein said synchronizer comprises the primary replica in systems based on a primary/backup replication strategy.

32. A method as recited in claim 30, wherein said replicas in said group compete to become said synchronizer for the given round in systems based on an active replication strategy.

33. A method as recited in claim 32:

wherein said replicas compete to win a given round in response to the order of delivering said clock synchronization message to said group of replicas;

wherein said replica whose clock synchronization message is delivered first to said group of replicas in a round of clock synchronization messages is the synchronizer that establishes the group clock value which is consistent across the group of replicas.

34. A method as recited in claim 33:

wherein during the initial clock synchronizing round, said group clock value is initialized to the local logical clock value of said synchronizer which is equivalent to the value of the physical hardware clock of said synchronizer;

wherein during subsequent clock synchronizing rounds, said group clock value is set to the local logical clock value of the synchronizer as the sum of its physical hardware clock value and its offset of the group clock from the local clock value in the previous round.

35. A method as recited in claim 12, wherein in response to delivering the message containing the local logical clock value proposed for the group clock to a non-faulty replica, it will be delivered to all non-faulty replicas.

36. A method as recited in claim 12, wherein in systems based on a primary/backup approach, failure of the primary replica before it sends said clock synchronization message, or failure of the primary replica after it sends said clock synchronization message but in which said clock synchronization message is not delivered to any non-faulty replica, the new primary replica sends a clock synchronization message.

37. A method as recited in claim 12, wherein said method is applicable to active replication, to both cold and warm passive replication, and to both semi-passive and semi-active replication strategies.

38. A method as recited in claim 12, wherein failure of a replica within said group of replicas does not interfere with the proposal and establishment of a group clock value by non-faulty replicas.

39. A method as recited in claim 12, wherein said method is implemented on top of a replication infrastructure and a group communication system.

40. A method as recited in claim 39, wherein said replication infrastructure and group communication system comprises a reliable ordered multicast protocol that ensures that replicas in said group receive the same messages in the same order, and that a message is delivered either to all of the replicas in said group or to

none of them.

41. A method as recited in claim 12, wherein said method is implemented utilizing library interpositioning of clock-related system calls to achieve application transparency.

42. A method of maintaining clock consistency for a group of replicas operating in a fault-tolerant distributed system, said clock consistency maintained as said replicas access their respective physical hardware clocks through a time service handler, comprising:

processing clock synchronization messages that arrive at a given replica from other replicas within said group of replicas;

updating a time offset value for said given replica as the difference between the physical hardware clock value for said given replica and a clock value contained in said clock synchronization message;

determining a local logical clock value from the sum of said physical hardware clock value and said time offset value;

proposing said local logical clock value by said given replica as a group clock value for replicas in said group of replicas by multicasting a clock synchronization message to the replicas in said group of replicas;

establishing said proposed clock value as a group clock value by other replicas in said group of replicas;

updating said clock offset value by each said replica accepting said group clock value; and

returning said group clock value to said given replica from said time service handler.

43. A method as recited in claim 42, wherein said clock synchronization message contains a proposed group clock value, a group identifier, and a sequence number of the message sent on the connection.